

Multi-User Dynamic Spectrum Access Based on LR-Q Deep Reinforcement Learning Network

Yuhui Li, Yu Wang, Yue Li, Bin Shen

School of Communication and Information Engineering(SCIE)

Chongqing University of Posts and Telecommunications(CQUPT), Chongqing 400-065, P. R. China

lyh12208@163.com, s200131057@stu.cqupt.edu.cn, ly2507321612@163.com, shenbin@cqupt.edu.cn

Abstract—Dynamic spectrum access (DSA) based spectrum sharing is an effective technique to alleviate the shortage of radio spectrum supply. Many DSA schemes based on deep reinforcement learning (DRL) have been proposed in the literature, but most of them merely dealt with simple heterogeneous cognitive radio networks (CRN). Combining two typical types of DRL networks, namely Long-short term memory (LSTM) and Residual network (ResNet), we proposes a distributed DSA algorithm for complex heterogeneous CRN. The algorithm aims to achieve high channel utilization and control the interference of SU to PU at a low level. The LSTM is used to extract the temporal characteristics of the historical spectrum data sequence to improve the spectrum state prediction capability. In contrast, the ResNet solves the performance degradation problem of deep neural networks caused by network depth and improves the training accuracy. Simulation results show that the algorithm can significantly improve spectrum utilization while controlling the interference of SU to PU.

Keywords— Dynamic spectrum access; Deep reinforcement learning; Heterogeneous; ResNet; LSTM

I. INTRODUCTION

According to Cisco's industry forecasts, the number of wireless devices connected to IP networks will reach 29.3 billion by 2023, and the demand for wireless networks will be 250 times higher than today^[1]. However, stringently limited wireless spectrum and static spectrum allocation cannot meet the growing demand for wireless spectrum^{[2][3]}. Aiming to solve the problem, DSA serves as an effective technology to improve the spectrum utilization efficiency by allowing the secondary user (SU) to dynamically discern the idle spectrum and gain opportunistically access. In order to avoid interference to the primary user (PU) as much as possible, the cognitive user continuously and actively monitors the spectrum for the PU presence.

In recent years, rapid development and excellent application of machine learning have provided new ideas for bringing DSA into feasible approaches^[4]. In [5], the authors proposed an access scheme for spectrum sharing between LTE-U and WiFi users through Q-learning, which is challenging to handle the large state space problems due to the limitations of Q-learning itself using table updates. After that, DRL with model-free features, especially DQN, was widely concerned, which uses deep neural networks to replace the Q-value table to fit the Q values and overcome the limitations of traditional Q-learning. In [6], the proposed DQN method outperforms all

previous algorithm and some human game in Atari. In [7][8], the author proposed a centralized spectrum allocation algorithm via DQN, but the central controller required in the network brings heavy channel control overhead to the underlying network and may reduce the security and stability of the network. The authors in [9][10] focused on combination of two spectrum access technologies under the game theory model, but the method relies on environmental knowledge, and the performance is constrained when the prior knowledge is insufficient. Based on deep recurrent neural network, a DQN method was proposed to solve the spectrum access problem between multiple channels and multiple PU^[11]. However, the authors only considered the problem of spectrum sharing between SU, and does not consider the existence of PU. The authors in [12] studied the spectrum sharing of the relevant channel mode, and the performance achieved using the DQN algorithm is close to the ideal performance of Myopic. However, the relevant channel mode will over-simplify the channel dynamics and the access strategy. In [13], the author proposed a distributed DRL spectrum access algorithm based on reservoirs computing. Although the algorithm achieves a high SU channel access rate, the interference of SU on PU was not dealt with. Similarly, the authors in [13] only studied the DSA problem of one kind PU in the network, and the algorithm implementation is of high complexity.

With an aim to tackle the multiple heterogeneous user coexisting problem, we propose a multi-user DSA algorithm for complex heterogeneous CRN in this paper. The DSA algorithm under consideration is based on Multi-User Deep Reinforcement Learning (MUDRL), and it combines both LSTM and ResNet. The proposed algorithm is referred to as Multi-user Deep Reinforcement learning Spectrum Access (MDRSA) algorithm. Unlike the previous spectrum access algorithm, MDRSA uses the neural network composed of LSTM and ResNet as the fitting Q network, which is called as the LR-Q network. In addition, the MDRSA algorithm will comprehensively consider the action results of PU and SU at the same time when making action selection, so as to further improve the agent's ability to predict the spectrum dynamics. Simulation results show that in the complex heterogeneous network environment, MDRSA can achieve higher channel utilization and is able to control the interference of SU to PU at a low level.

The rest of this paper is organized as follows. In Section II,

the system model is described. Section III presents the proposed MDRSA based on Multi-User Reinforcement Learning. In Section IV, the performance evaluation results of the proposed algorithm are given. Section V concludes the paper.

II. SYSTEM MODEL

Consider a distributed heterogeneous cognitive radio network, where there are N SUs. In addition, there are M PUs whose MAC protocols are not identical. The purpose of this setting is to verify that SU can share the licensed spectrum of multiple heterogeneous PUs. Suppose that each PU orthogonally occupies only one OFDM channel. We use an interweave sharing model^[14], in which the SU monitors the state of the licensed channel and accesses the idle channel in a timely manner, where only when the PU does not use the licensed channel can the SU attempt to access the channel for transmission. The model in this paper aims to describe a scenario where each SU accesses an licensed channel using interweave policies in a distributed cognitive radio heterogeneous network, where multiple SUs compete for one transmission access point.

According to the different MAC protocols executed by different users in the realistic environment, we consider two different heterogeneous wireless network PU nodes. One is a time-slot node using time-division multiplexing, and the other is the probability node that accesses the channel with a certain probability p^* in each time slot. We assume that SUs always have transmission requirements. Therefore, the SU will attempt to access the channel whenever a spectrum hole is detected.

As shown in Figure 1, multiple PU and SU are randomly distributed across the network, with different PUs using different licensed channels and the SUs maintaining channel state awareness. In the figure, $S-T$ represents the transmitter of a time-slot PU node, and its corresponding receiver is $S-R$; $P-T$ represents the transmitter of a probability PU node and its corresponding receiver is $P-R$. $SU-T$ represents the transmitter of a SU node and $SU-R$ denotes its corresponding receiver. In our research scenario, there are multiple dynamic channels. Figure 1 shows the time-slot channel and probability channel. The dynamic model of these two channels is shown by one example in Figure 2. In Figure 2, channel 1 and channel 3 show the dynamic rule of time-slot channel, which can be represented as (1):

$$s(a) = \begin{cases} 1 & t-C < t+a-C \leq t, a \in \{i, j, k\} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $s(a)=1$ indicates that the channel is occupied and $s(a)=0$ indicates that the channel is idle. In the case of Channel 1, the time-slot user will occupy up to X ($X \leq C$) of the continuous C time slots, then there are 2^X occupancy forms. For example, a time-slot user occupies the i^{th} , j^{th} , and k^{th} slot for transmission, while any SU is prohibited from accessing the channel during these time-slots. In Figure 2, Channel 2 shows the dynamic rule of the probability channel. In each

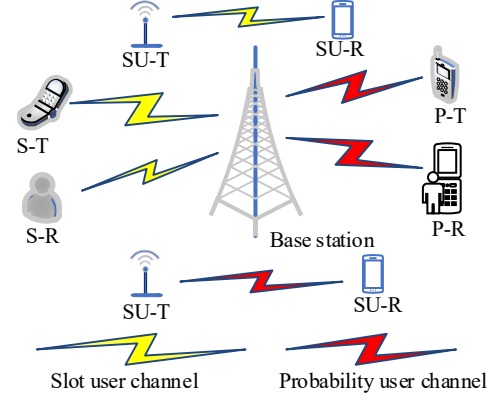


Fig. 1. System model

time-slot, the probability user occupies that time slot for transmission. The dynamic rule of the channel can be expressed as (2):

$$s(a) = \begin{cases} 1 & p < p^* \\ 0 & p > p^* \end{cases} \quad (2)$$

Specifically, the probability user will generate a random number $p \in [0,1]$ at the beginning of each time slot, and if $p < p^*$, the probability user will occupy the channel for transmission. If $p > p^*$, the probability user remains silent and does not occupy the channel. In this case, the licensed channel remains idle and the SU can access the channel. Assuming that each of the above channels is independent of each other, we model each channel as a two-state Markov model, that is, each channel has two states of idle (1) and occupied (0), and only one state can exist for the one time-slot channel.

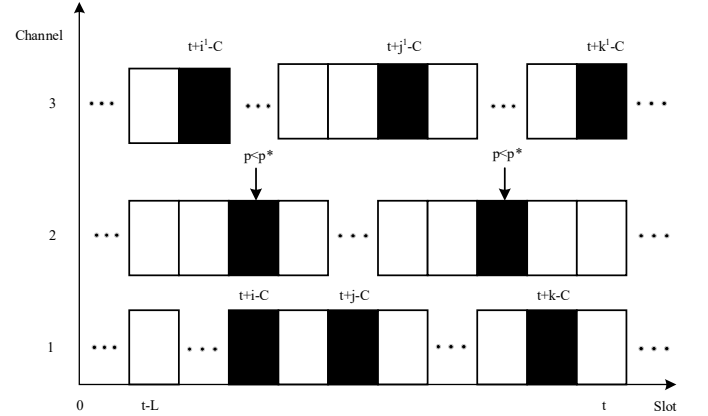


Fig. 2. Multichannel dynamic pattern

Based on the above system model and with an aim to improve network channel utilization and control interference, the SU senses the channel state at the beginning of each time slot and decides what actions to take next. The SUs will observe the effect of actions on completion of specified tasks from the environment. The measured indicator is reward, SU gets corresponding rewards by observing the results after the action is performed. If one SU chooses not to transmit, the reward signal of the environmental feedback is R_i , which

returns the channel state observation signal at this time. If the SU chooses to access a channel, the environment will feedback positive reward R_s or negative reward R_f . We use Reinforcement Learning (RL) to maximize long-term expected rewards, maximize channel utilization and reduce spectrum access conflicts, namely:

$$E = \max_a Q(s, a) \quad (3)$$

Where $Q(s, a)$ is the output function of the standard Q-learning, indicating the value of executing action a in state s . In Q-learning, it is updated by the following:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (4)$$

Where s_t is the current state, s_{t+1} is the next moment state, a_t is the current action, a_{t+1} is the next moment action, r_{t+1} is the delay reward, $\alpha \in [0, 1]$ is the learning rate, $\gamma \in [0, 1]$ is the reward discount, which weighs the importance of immediate reward and future reward. In the following work, with Q-learning, the reward is defined as a parameter that is positively related to successful access and negatively related to access failure. Then maximizing the licensed channel utilization and reducing conflicts is made possible by maximizing the expected cumulative reward.

III. SPECTRUM ACCESS STRATEGY FOR COMPLEX HETEROGENEOUS NETWORKS

A. Multi User Reinforcement Learning

At present, the proposed DSA algorithm based on machine learning only directly considers the channel state, and does not specifically study the impact of each SU on the channel state. Based on the work in [15], we propose a MUDRL based multiuser spectrum access algorithm, which considers the impact of each SU on the channel state.

Different from the traditional RL, our MUDRL quantifies the state action results of each PU, and then takes them into consideration in the action decision-making of agents. After performing the optimal action at each time step, the agent observes a reward vector in the environment. Each element in the reward vector corresponds to a user node, which is used to update the user's state action value $Q^{(i)}(s, a)$. Since the agent can only perform a single action at any time, in order to consider the impact of PU access results on the agent's action decisions, we design a comprehensive objective function $Q_T(s, a)$ for MUDRL, which integrates the single objective function $Q^{(i)}(s, a)$ of each user node, representing the amount of impact of all user nodes on the agent's action decisions. By comprehensive objective function $Q_T(s, a)$, the agent can comprehensively consider the impact of the access results of all nodes on the agent's action decisions, so as to select the optimal target action^[16].

Assuming that there are L heterogeneous PUs in the network, then there are L orthogonal channels. The comprehensive objective function of each agent can be defined as:

$$Q_T(s, a) = \sum_{i=1}^{L+1} w_i Q^{(i)}(s, a) \quad (5)$$

where $i = 1, 2, \dots, L+1$, $w_i \in [0, 1]$ is the weight coefficient, and $Q^{(i)}(s, a)$ is the action value function of the node i . Where $i = 1, 2, \dots, L$ indicates that the node is a PU node, and $i = L+1$ indicates that the node is an agent node. After the agent executes the action, the agent will observe the effect of the action in the environment on completing the task, and generate a $L+1$ dimension reward vector. Each element in the reward vector represents the transmission result of the corresponding node, and the reward will be used for update the value function $Q(s, a)$ of the state action pair (s, a) of the corresponding node. Specifically, let $r^{(i)}$ represent the reward corresponding to the node, so the reward vector received by the agent can be expressed as $[r^{(i)}]_{i=1}^{L+1}$. Therefore, for a state action pair (s, a) , MUDRL agents no longer maintain the scalar value $Q(s, a)$, but need to maintain a value vector $[Q^{(i)}(s, a)]_{i=1}^{L+1}$, where $Q^{(i)}(s, a)$ represents the cumulative discount reward of node i .

Unlike traditional Q-learning, which uses table to update the Q value of the maintenance state action pair, MUDRL uses neural networks to fit the value function $Q(s, a)$ of the state action pair (s, a) . In the phase of neural network fitting $Q(s, a)$, the algorithm uses $q(s, a)$ to represent the fitting value corresponding to $Q(s, a)$ of the state action pair (s, a) , and the corresponding comprehensive objective function is expressed as $q_T(s, a)$. Assume that at time t , the environment state is s_t , and the algorithm uses ϵ -greedy policy to make decisions when selecting actions, using the following formula:

$$a_t = \begin{cases} \arg \max_a q_T(s_t, a) & \text{with } 1 - \epsilon \\ \text{random action} & \text{with } \epsilon \end{cases} \quad (6)$$

The reason for randomly selecting actions is to avoid the local optimization of agent due to $q(s, a)$ that does not converge to $Q^*(s, a)$. Where $q_T(s_t, a) = \sum_{i=1}^{L+1} q^{(i)}(s_t, a)$. After executing the action a_t , the agent will simultaneously update each element $q^{(i)}(s_t, a)$, $i = 1, 2, \dots, L+1$, in the $L+1$ dimension fitting Q value vector, that is

$$q^{(i)}(s_t, a_t) \leftarrow q^{(i)}(s_t, a_t) + \alpha[r_{t+1}^{(i)} + \gamma q^{(i)}(s_{t+1}, a_{t+1}) - q^{(i)}(s_t, a_t)] \quad (7)$$

where,

$$q_T(s_{t+1}, a') = \sum_{i=1}^{L+1} q^{(i)}(s_{t+1}, a') \quad (8)$$

$$a_{t+1} = \arg \max_a q_T(s_{t+1}, a') \quad (9)$$

Note that the update of MUDRL is different from that of traditional Q-learning. In traditional Q-learning, actions with the optimal Q-function are selected. However, in MUDRL, the algorithm uses a double Q network to decouple the direct relationship between the target action selection and the optimal

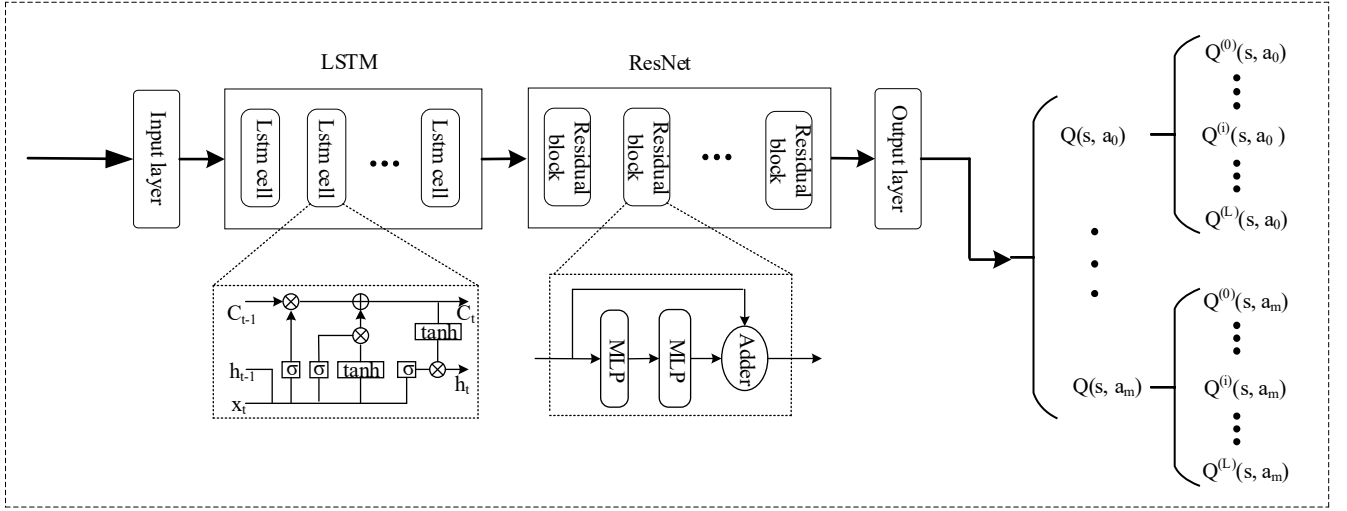


Fig. 3. MDRSA agent architecture

Q function. Secondly, in MUDRL, in order to consider the impact of the access result of the PU on the agent's action decision, when selecting the target action, we use a comprehensive objective function. Therefore, from the above analysis, when making action decisions, MUDRL no longer only considers the value function $Q(s, a)$ of the agent itself, but also considers a comprehensive objective function $Q_T(s, a)$ that integrates the value functions of multiple users. $Q_T(s, a)$ directly considers the impact of the access results of the PUs on the agent's action choices, thus improving the accuracy of the agent's action decisions.

B. MDRSA Algorithm Based on Multi-user Reinforcement Learning

On the basis of MUDRL, we propose a dynamic spectrum access algorithm for complex heterogeneous radio networks, namely MDRSA algorithm. Similar to traditional RL, MUDRL is also a modelless learning method. It gradually summarizes historical experience from the interaction with the environment and learns action decision-making policy. The MDRSA algorithm aims to improve the channel utilization and control the interference of SU to PU at a low level. Figure 3 shows the basic structure of MDRSA algorithm agent.

As mentioned above, M heterogeneous PUs and N SUs coexist in a distributed cognitive radio heterogeneous network. Each SU is considered as an agent, and each agent interacts with the environment independently to learn decision-making policy. At the beginning of each slot, the agent's perception of the channel generates a state vector with a length of M , which is recorded as o_t . For example, when the agent senses channel 2 is occupied, the state vector $o_t = \{0, 1, \dots, 0\}$. At time t , it is assumed that the environment input state of the agent is represented as s_t , and the action selected by the agent is $a_{mt} = \{0, \dots, a_t, \dots, 0\}$, indicating that the action of the agent acts on the m^{th} channel, where $a_t \in \{transmit, wait\}$, *transmit* indicates that the agent selects the channel for transmission,

and *wait* indicates that the agent does not transmit. After the agent executes the action, the agent will observe the result $z_{mt} \in \{successful, failed, idle\}$ of the action on the channel state in the environment, where *idle* indicates the channel is idle, *successful* indicates successful data transmission, and *failed* indicates access failure. Therefore, we define the channel state at time $t+1$ as $c_{t+1} = (a_{mt}, o_t, z_{mt})$. In addition, we define the environmental state at time $t+1$ as $s_{t+1} = [c_{t-H+2}, \dots, c_t, c_{t+1}]$, where H is the length of historical state to be considered by the agent. At the same time, the environment state is also the input state of the agent. When the agent executes action a_{mt} , the PU will also access the channel according to its own transmission requirements, so the environment state will be transferred from s_t to s_{t+1} , and the agent will observe a reward vector $[r_{t+1}^{(i)}]_{i=1}^{L+1}$. The $r_{t+1}^{(i)}$ reward function is defined as follows:

$$r_{t+1}^{(i)} = \begin{cases} R_s & \text{if } z_i = \text{successful} \\ R_i & \text{if } z_i = \text{idle} \\ R_f & \text{if } z_i = \text{failed} \end{cases} \quad (10)$$

where R_s indicates that the user has successfully occupied the channel for transmission; R_i indicates that the user remains idle at this time; R_f indicates that the user failed to access the channel. When the SU fails to access, the agent will learn the experience of the failed access, update the access policy, and avoid the recurrence of such events in subsequent timeslots. Then, the agent combines the experience generated at the last moment into experience tuple $(s_t, a_{mt}, r_{t+1}, s_{t+1})$ and stores it in the experience pool. When the number of experiences reaches a certain value, the agent starts to extract samples from the experience pool and perform experience replay. Because MDRSA uses double Q network, the agent updates the network parameters of the target network every F times of training. The pseudo code of the above algorithm is shown in Algorithm 1.

Algorithm 1 MDRSA	
Input: Environment state vector s_t	
Output: Network parameters θ, θ^- ; policy π^*	
1: Initialize $s_0, \varepsilon, \alpha, \gamma, F$	
2: Initialize experience pool <i>Memory</i> , <i>mini-batch</i>	
3: Initialize estimated network parameter is θ	
4: Initialize target network parameter is θ^-	
5: For $t = 0, 1, 2, \dots, T$ do	
6: input s_t , output $Q = \{q(s_t, a, \theta) a \in A\}$	
7: execute ε -greedy and generate action a_{mt}	
8: the agent executes action a_{mt} and observes $z_{mt}, \mathbf{r}_{t+1}$	
9: calculate s_{t+1} according to s_t, a_{mt}, z_{mt}, o_t	
10: storage $(s_t, a_{mt}, \mathbf{r}_{t+1}, s_{t+1})$ into <i>Memory</i>	
11: $s_t = s_{t+1}$	
12: If $t > 100$ and $t\%5 = 0$ do	
13: sample experience from <i>Memory</i> and perform experience replay	
14: End if	
15: If $t > 100$ and $t\%F = 0$ do	
16: extract θ to update θ^- of target network	
17: End if	
18: End for	

IV. SIMULATIONS

In this section, we verify the performance of the algorithm through the Tensorflow platform. The traditional Q-learning algorithm and DQN algorithm, as the comparison algorithm of our MDRSA, have been simulated and compared in detail.

A. Simulation Settings

The performance indicators in the simulation are the utilization of idle timeslot channels by SUs, the utilization of all channels by network users, and the interference rate of SU to PU. The channel dynamics of heterogeneous users used in the simulation are as follows:

1) **Time slot node:** Each slot node occupies the i^{th} , j^{th} and k^{th} slots of the consecutive C slots for transmission.

2) **Probability node:** the probability threshold of the probability node accessing the channel transmission in each time slot is p^* .

Other parameter settings in the experiment are shown in Table I.

TABLE I

HYPERPARAMETERS OF MDRSA	
Hyperparameters	Values
T	15000
F	20
ε	$1.0 \rightarrow 0.005$

γ	0.9
α	0.01
H	10
Input layer size	7*280
Output layer size	1*1

B. Experimental Results and Discussions

In this scenario, we simulate a multi heterogeneous multi-user network with four slot node users, one probability node user, and seven SUs. This scenario is mainly to prove that MDRSA has good performance in the complex heterogeneous multi-user network environment. From Figure 4, the MDRSA achieves 83% utilization of idle timeslot channels by SUs under multiple heterogeneous channels, achieving high utilization of idle timeslot channels. DQN and traditional Q-learning achieve 74% and 40% idle slot channel utilization respectively. From Figure 5, in the multi heterogeneous channels environment, the MDRSA achieves 87.6% utilization of all channels, achieving high utilization of channels. The DQN algorithm achieves 77% channel utilization, while the traditional Q-learning algorithm only achieves 56% channel utilization. Let's assume $C=10$, $p^*=0.2$, where there are only four slot nodes and one probability node, the optimal channel utilization is 28%. We add 7 SU pairs to the network, the MDRSA increases the channel utilization by 3.13 times, the DQN algorithm increases the channel utilization by 2.75 times, and the traditional Q-learning algorithm increases the channel utilization by 2 times. Figure 6 shows the PU interference when the algorithm training converges. The white square indicates that the PU is disturbed, and the black square indicates that there is no conflict. From Figure 6, MDRSA algorithm has only minor interference to the PUs, DQN algorithm has more interference to the PUs, while traditional Q-learning algorithm has serious interference to the PUs.

To sum up, in the multi heterogeneous multi-user scenario, although DQN and Q-learning both improve channel utilization, they have a large interference to the PUs, especially the interference of Q-learning to the PUs, which seriously affects normal communication of the PUs. On the contrary, our proposed MDRSA not only improves the channel utilization, but also has little interference to the PUs, which does not affect its normal communication.

V. CONCLUSIONS

We studied the distributed dynamic spectrum access problem of complex heterogeneous networks in cognitive radio. Based on MUDRL, combined with LSTM and ResNet, we proposed the MDRSA spectrum access algorithm. In the previous article, we introduced the structure, principle and implementation of MDRSA in detail. The purpose of MDRSA is to achieve high channel utilization of SU and low interference to PU in a complex heterogeneous network. Finally, we verified the feasibility of the MDRSA spectrum access algorithm in a multi heterogeneous multi-user scenario.

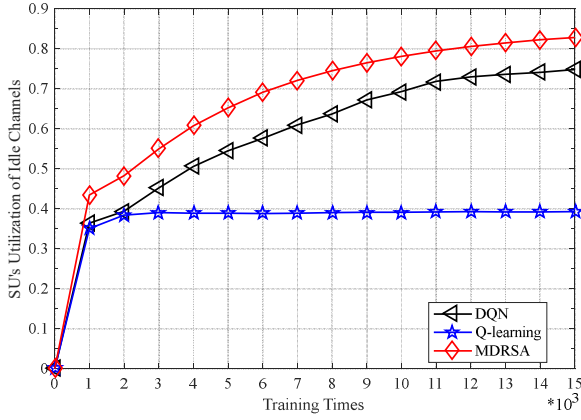


Fig. 4. SU utilization of idle slot channels

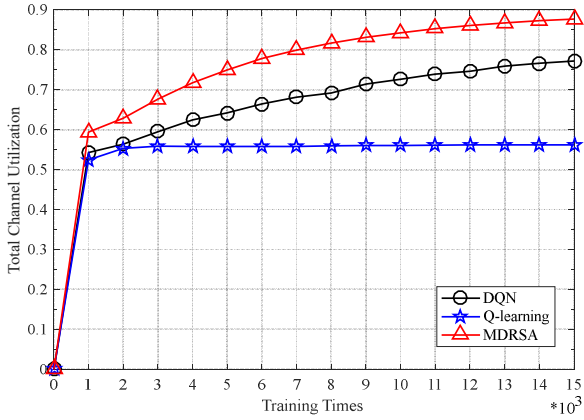


Fig. 5. Total channel utilization for complex heterogeneous networks

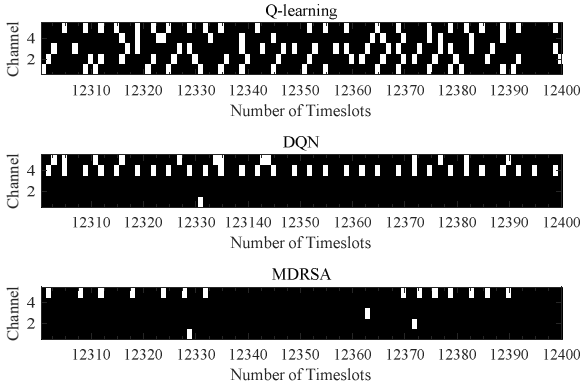


Fig. 6. Interference diagram of complex heterogeneous network SU to PU

Compared with DQN and traditional Q-learning, MDRSA spectrum access algorithm has better access performance. MDRSA can greatly improve the utilization rate of the channel, and only cause minor interference to the network PU, without affecting its normal communication.

ACKNOWLEDGMENT

This work was supported by The National Natural Science Foundation of China under Grant 62071078.

REFERENCES

- [1] U. Cisco. "Cisco annual internet report (2018–2023) white paper. 2020". *Acessado em*, 10(01). 2021
- [2] K. Chen, J. Min, X. Han, et al. "Spectrum survey for TV band in Beijing", *2014 21st International Conference on Telecommunications (ICT)*, 2014, paper 267-271.
- [3] V. Dhivya, P. Kalaiyarasi, G.I. Shamini. "Survey on spectrum occupancy by using different techniques", *2017 International Conference on Computation of Power, Energy Information and Commuication (ICCPEIC)*, 2017, paper 316-320.
- [4] Y. Liang, J. Tan, D. Niyato. "Overview on intelligent wireless communication technology". *Journal on Communications*, 41(7):pp. 1-17. July 2020
- [5] N. Rastegardoost, B. Jabbari. "A Machine Learning Algorithm for Unlicensed LTE and WiFi Spectrum Sharing", *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. 2018, paper 1-6.
- [6] U. Kaytaz, S. Ucar, B. Akgun, et al. "Distributed Deep Reinforcement Learning with Wideband Sensing for Dynamic Spectrum Access", *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. 2020, paper 1-6.
- [7] Q. Cong, W. Lang. "Deep Multi-User Reinforcement Learning for Centralized Dynamic Multichannel Access", *2021 6th International Conference on Intelligent Computing and Signal Processing (ICSP)*. 2021, paper 824-827.
- [8] Z. Shi, X. Xie, H. Lu, et al. "Deep-Reinforcement-Learning-Based Spectrum Resource Management for Industrial Internet of Things". *IEEE Internet of Things Journal*, 8(5):pp. 3476-3489. September 2021.
- [9] O. Naparstek, K. Cohen. "Deep Multi-User Reinforcement Learning for Distributed Dynamic Spectrum Access". *IEEE Transactions on Wireless Communications*, 18(1):pp. 310-323. November 2019.
- [10] S. Wang, H. Liu, P.H. Gomes, et al. "Deep Reinforcement Learning for Dynamic Multichannel Access in Wireless Networks". *IEEE Transactions on Cognitive Communications and Networking*, 4(2):pp. 257-265. February 2018.
- [11] H.H. Chang, H. Song, Y. Yi, et al. "Distributive Dynamic Spectrum Access Through Deep Reinforcement Learning: A Reservoir Computing-Based Approach". *IEEE Internet of Things Journal*, 6(2):pp. 1938-1948. April 2019.
- [12] N. Morozs, D. Grace, T. Clarke. "Distributed Q-learning based dynamic spectrum access in high capacity density cognitive cellular systems using secondary LTE spectrum sharing", *2014 International Symposium on Wireless Personal Multimedia Communications (WPMC)*. 2014, paper 462-467.
- [13] F. Shah-Mohammadi, A. Kwasinski. "Deep Reinforcement Learning Approach to QoE-Driven Resource Allocation for Spectrum Underlay in Cognitive Radio Networks", *2018 IEEE International Conference on Communications Workshops (ICC Workshops)*. 2018, paper 1-6.
- [14] F.Z. J. XU, K. LI, Y. LI. "Performance analysis of two-way relay cooperation underlay cognitive radio networks based on WIPT". *Journal on Communications*, 40(2):pp. 129-136. September 2019.
- [15] K. Li, T. Zhang and R. Wang, "Deep Reinforcement Learning for Multiobjective Optimization," in *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3103-3114, June 2021.
- [16] C. Liu, X. Xu, D. Hu. "Multiobjective Reinforcement Learning: A Comprehensive Overview". *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(3):pp. 385-398. March 2015.



Yuhui Li (1997-) is a master's student from Chongqing University of Posts and Telecommunications, with research interests in deep reinforcement learning, cognitive radio and dynamic spectrum access.